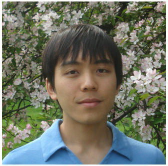
Brown clusters, linguistic context, and spectral algorithms

Daniel Hsu

Columbia University

Joint work with



Mike Collins (Columbia/Google)

Do-kyum Kim (Google)

Karl Stratos (Columbia)

1. Introduction

Learning from unlabeled data

Many applications of machine learning

- Lots of high-dimensional data.
- Mostly unlabeled—i.e., not annotated with prediction target.

Learning from unlabeled data

Many applications of machine learning

- Lots of high-dimensional data.
- Mostly unlabeled—i.e., not annotated with prediction target.

What kinds of structure can we learn from unlabeled data?

Examples from natural language processing

- **Example 1**: Language models

$$\frac{P(\text{colorless green ideas sleep furiously})}{P(\text{furiously sleep ideas green colorless})} \gg 1$$

Examples from natural language processing

- **Example 1**: Language models

$$\frac{P(\text{colorless green ideas sleep furiously})}{P(\text{furiously sleep ideas green colorless})} \gg 1$$

- **Example 2**: Word sense disambiguation

$$\begin{aligned} & \big(\text{``bank''}, \{\text{``stocks''}, \text{``bonds''}, \dots\}\big) \\ \text{vs.} \quad & \big(\text{``bank''}, \{\text{``river''}, \text{``freshwater''}, \dots\}\big) \end{aligned}$$

Examples from natural language processing

- **Example 1**: Language models

$$\frac{P(\text{colorless green ideas sleep furiously})}{P(\text{furiously sleep ideas green colorless})} \gg 1$$

- **Example 2**: Word sense disambiguation

$$\begin{aligned} & \big(\text{``bank''}, \{\text{``stocks''}, \text{``bonds''}, \dots\}\big) \\ \text{vs.} \quad & \big(\text{``bank''}, \{\text{``river''}, \text{``freshwater''}, \dots\}\big) \end{aligned}$$

Doesn't require any direct supervision to learn!

Examples from natural language processing

- **Example 1**: Language models

$$\frac{P(\text{colorless green ideas sleep furiously})}{P(\text{furiously sleep ideas green colorless})} \gg 1$$

- **Example 2**: Word sense disambiguation

$$\begin{aligned} & \big(\text{``bank''}, \{\text{``stocks''}, \text{``bonds''}, \dots\}\big) \\ \text{vs.} \quad & \big(\text{``bank''}, \{\text{``river''}, \text{``freshwater''}, \dots\}\big) \end{aligned}$$

- **Example 3**: "Word classes"

$$\begin{aligned} \text{e.g., } & \{\text{``apple''}, \text{``pear''}, \dots\},\ \{\text{``Apple''}, \text{``IBM''}, \dots\}, \\ & \{\text{``bought''}, \text{``run''}, \dots\},\ \{\text{``of''}, \text{``in''}, \dots\}, \dots \end{aligned}$$

Doesn't require any direct supervision to learn!

Word class models

Brown, Della Pietra, deSouza, Lai, and Mercer (1992): "Class based *n*-gram models of natural language"

- **Brown clustering**: clustering a vocabulary into word classes using the Brown clustering algorithm

class 1	class 2	class 3	⋯
feet	people	water	
miles	guys	gas	
pounds	folks	coal	
degrees	fellows	liquid	
⋮	⋮	⋮	

Word class models

Brown, Della Pietra, deSouza, Lai, and Mercer (1992): "Class based *n*-gram models of natural language"

- **Brown clustering**: clustering a vocabulary into word classes using the Brown clustering algorithm

class 1	class 2	class 3	…
feet	people	water	
miles	guys	gas	
pounds	folks	coal	
degrees	fellows	liquid	
⋮	⋮	⋮	

Q: What do these word classes capture?

Word class models

Brown, Della Pietra, deSouza, Lai, and Mercer (1992): "Class based *n*-gram models of natural language"

- **Brown clustering**: clustering a vocabulary into word classes using the Brown clustering algorithm

class 1	class 2	class 3	...
feet	people	water	
miles	guys	gas	
pounds	folks	coal	
degrees	fellows	liquid	
⋮	⋮	⋮	

Q: What do these word classes capture?

Not entirely clear, but ...

Structure from a language model

Semi-supervised Natural Language Processing:

Structure from a language model

Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive "lexical representations" (a.k.a. word representations).

Structure from a language model

Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive “lexical representations” (a.k.a. word representations).
2. Augment existing NLP methods with lexical representations.

Structure from a language model

Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive "lexical representations" (a.k.a. word representations).
2. Augment existing NLP methods with lexical representations.
3. Win!
   - Named-entity recognition (Miller *et al*, 2004; Turian *et al*, 2010)
   - Dependency parsing (Koo *et al*, 2008)
   - Language modeling* (Kneser and Ney, 1993; Gao *et al*, 2001)
   - ...

Structure from a language model

Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive "lexical representations" (a.k.a. word representations).
2. Augment existing NLP methods with lexical representations.
3. Win!
   - Named-entity recognition (Miller *et al*, 2004; Turian *et al*, 2010)
   - Dependency parsing (Koo *et al*, 2008)
   - Language modeling* (Kneser and Ney, 1993; Gao *et al*, 2001)
   - ...

Our goal: Understand & build on the success of Brown clustering

Our contributions

Motivating observation: Learning Brown word classes only requires correlations between words & simple linguistic context.

Our contributions

Motivating observation: Learning Brown word classes only requires correlations between words & simple linguistic context.

What we do:

1. Propose a spectral algorithm for learning word classes in the setting of Brown *et al* [Stratos, Kim, Collins, & H, UAI 2014]
   - Algorithmically simple, amenable to theoretical analysis
   - Empirically faster than Brown clustering algorithm

Our contributions

Motivating observation: Learning Brown word classes only requires correlations between words & simple linguistic context.

What we do:

1. Propose a spectral algorithm for learning word classes in the setting of Brown *et al* [Stratos, Kim, Collins, & H, UAI 2014]
   - Algorithmically simple, amenable to theoretical analysis
   - Empirically faster than Brown clustering algorithm
2. Address noise heteroskedasticity using variance stabilization [Stratos, Collins, & H, ACL 2015]
   - Theoretically understood in Brown *et al* setting
   - Improves lexical representations for low-level NLP tasks

Our contributions

Motivating observation: Learning Brown word classes only requires correlations between words & simple linguistic context.

What we do:

1. Propose a spectral algorithm for learning word classes in the setting of Brown *et al* [Stratos, Kim, Collins, & H, UAI 2014]
   - Algorithmically simple, amenable to theoretical analysis
   - Empirically faster than Brown clustering algorithm
2. Address noise heteroskedasticity using variance stabilization [Stratos, Collins, & H, ACL 2015]
   - Theoretically understood in Brown *et al* setting
   - Improves lexical representations for low-level NLP tasks
3. Assess ability of Brown word class model to capture real linguistic structure—real test of *unsupervised learning*. [Stratos, Collins, & H, TACL 2016]

Talk outline

1. Spectral algorithm for learning word classes in the setting of Brown *et al*
   [Stratos, Kim, Collins, & H, UAI 2014]

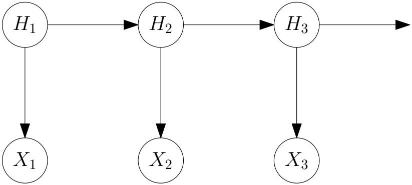
2. Improved estimation using variance stabilization
   [Stratos, Collins, & H, ACL 2015]

3. Using Brown word class model for unsupervised POS tagging
   [Stratos, Collins, & H, TACL 2016]

2. Examining the Brown word class model

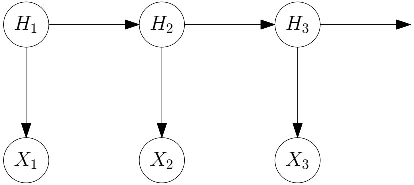
The Brown *et al* word class model (parameters)

HMM with hidden state seq. (H_t) and observation seq. (X_t).



The Brown *et al* word class model (parameters)

HMM with hidden state seq. (H_t) and observation seq. (X_t).



Hidden state space $=$ word classes $C := \{1, 2, \ldots, |C|\}$.

Observation space $=$ vocabulary $V := \{1, 2, \ldots, |V|\}$.

Column-stochastic parameters $\boldsymbol{\theta} := (\boldsymbol{\pi}, \boldsymbol{T}, \boldsymbol{O})$

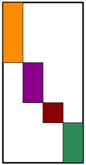
$$\begin{aligned}
\pi_h &= P_{\boldsymbol{\theta}}[H_1 = h]\,, && h \in C\,,\\
T_{g,h} &= P_{\boldsymbol{\theta}}\big[H_{t+1} = g \mid H_t = h\big]\,, && (g,h) \in C \times C\,,\\
O_{x,h} &= P_{\boldsymbol{\theta}}\big[X_t = x \mid H_t = h\big]\,, && (x,h) \in V \times C\,.
\end{aligned}$$

The Brown *et al* word class model (structural restriction)

Brown *et al* word class model places structural restriction on $\boldsymbol{O}$:

> *There is a hard clustering of vocabulary V into $|C|$ groups $\{V_h : h \in C\}$ (the word classes) such that*
>
> $$x \in V_h \quad \Longrightarrow \quad P_{\boldsymbol{\theta}}\big[X_t = x \mid H_t = g\big] \;=\; 0 \text{ for all } g \neq h\,.$$

Each word can be generated by the hidden state corresponding to its word class.



Sparsity pattern of
emission probablity matrix $\boldsymbol{O}$

(after permuting rows)

Log-likelihood in the word class model

Max-likelihood parameters that respect clustering $\mathcal{C}$ is (up to consts.)
empirical mutual informaton bet. word classes of adjacent words

$$\sum_t \sum_{g,h} \widehat{\Pr}\big[\mathcal{C}(X_t) = g, \mathcal{C}(X_{t+1}) = h\big] \ln \frac{\widehat{\Pr}\big[\mathcal{C}(X_t) = g, \mathcal{C}(X_{t+1}) = h\big]}{\widehat{\Pr}\big[\mathcal{C}(X_t) = g\big] \widehat{\Pr}\big[\mathcal{C}(X_{t+1}) = h\big]} .$$

Under the Brown word class model:
max log-likelihood $\Leftrightarrow$ max $\widehat{\text{MI}}$s between classes of adjacent words

Log-likelihood in the word class model

Max-likelihood parameters that respect clustering $\mathcal{C}$ is (up to consts.)
empirical mutual informaton bet. word classes of adjacent words

$$\sum_t \sum_{g,h} \widehat{\Pr}\left[\mathcal{C}(X_t) = g, \mathcal{C}(X_{t+1}) = h\right] \ln \frac{\widehat{\Pr}\left[\mathcal{C}(X_t) = g, \mathcal{C}(X_{t+1}) = h\right]}{\widehat{\Pr}\left[\mathcal{C}(X_t) = g\right] \widehat{\Pr}\left[\mathcal{C}(X_{t+1}) = h\right]} .$$

Under the Brown word class model:
max log-likelihood $\Leftrightarrow$ max $\widehat{\text{MI}}$s between classes of adjacent words

Not clear how to efficiently maximize w.r.t. clustering $\mathcal{C}$.

Log-likelihood in the word class model

Max-likelihood parameters that respect clustering $\mathcal{C}$ is (up to consts.)
empirical mutual informaton bet. word classes of adjacent words

$$\sum_t \sum_{g,h} \widehat{\Pr}\big[\mathcal{C}(X_t) = g, \mathcal{C}(X_{t+1}) = h\big] \ln \frac{\widehat{\Pr}\big[\mathcal{C}(X_t) = g, \mathcal{C}(X_{t+1}) = h\big]}{\widehat{\Pr}\big[\mathcal{C}(X_t) = g\big] \widehat{\Pr}\big[\mathcal{C}(X_{t+1}) = h\big]} .$$

Under the Brown word class model:
max log-likelihood $\Leftrightarrow$ max $\widehat{\text{MI}}$s between classes of adjacent words

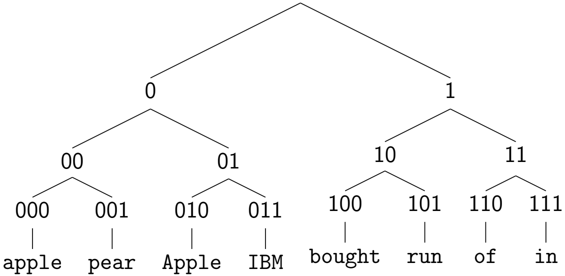
Not clear how to efficiently maximize w.r.t. clustering $\mathcal{C}$.

Brown clustering algorithm (Brown *et al*, 1992):

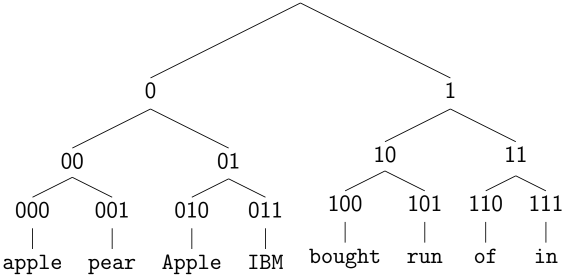
- Start with each word in its own class.
- Repeat: merge class pair that decreases $\widehat{\text{MI}}$s the least.

Output: a *hierarchy* of word classes.

Output of Brown clustering algorithm



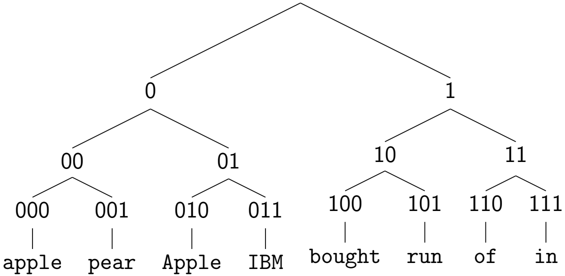
Output of Brown clustering algorithm



Get lexical representations from a pruning of the hierarchy:

apple	$\rightarrow$	00	bought	$\rightarrow$	10
pear	$\rightarrow$	00	run	$\rightarrow$	10
Apple	$\rightarrow$	01	of	$\rightarrow$	11
IBM	$\rightarrow$	01	it	$\rightarrow$	11

Output of Brown clustering algorithm



Get lexical representations from a pruning of the hierarchy:

apple	→	00	bought	→	10
pear	→	00	run	→	10
Apple	→	01	of	→	11
IBM	→	01	it	→	11

Use in NLP: augmenting text data with lexical representations increases ability for (supervised) ML methods to learn other linguistic structure.

Word classes from observable quantities

Our aim: extract word classes directly from observable quantities.

Word classes from observable quantities

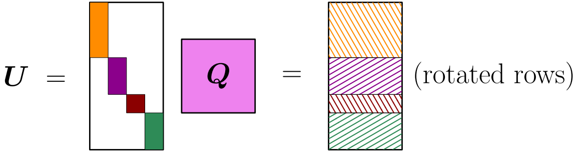
Our aim: extract word classes directly from observable quantities.

Theorem (Stratos, Kim, Collins, and H, 2014)

Define matrix $\boldsymbol{B} \in \mathbb{R}^{V \times V}$

$$B_{x,y} \;:=\; \sum_{t=1}^{n-1} P_{\boldsymbol{\theta}}(X_t = x,\, X_{t+1} = y)\,.$$

If data follow a Brown model distribution, then left singular vectors of $\boldsymbol{B}$ *"reveal" the word classes (after row normalization).*



Word classes from observable quantities

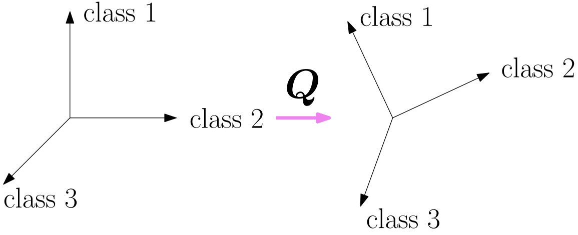
Our aim: extract word classes directly from observable quantities.

Theorem (Stratos, Kim, Collins, and H, 2014)

Define matrix $\boldsymbol{B} \in \mathbb{R}^{V \times V}$

$$B_{x,y} \ := \ \sum_{t=1}^{n-1} P_{\boldsymbol{\theta}}(X_t = x, X_{t+1} = y)\,.$$

If data follow a Brown model distribution, then left singular vectors of $\boldsymbol{B}$ *"reveal" the word classes (after row normalization).*



Word classes from observable quantities

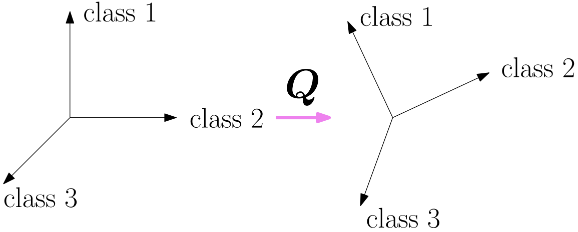
Our aim: extract word classes directly from observable quantities.

Theorem (Stratos, Kim, Collins, and H, 2014)

Define matrix $\boldsymbol{B} \in \mathbb{R}^{V \times V}$

$$B_{x,y} \ := \ \sum_{t=1}^{n-1} P_{\boldsymbol{\theta}}(X_t = x, X_{t+1} = y)\,.$$

If data follow a Brown model distribution, then left singular vectors of $\boldsymbol{B}$ "reveal" the word classes (after row normalization).



$\boldsymbol{B}$ can be estimated directly from raw collection of sentences.

Spectral algorithm for Brown clustering

Spectral algorithm for Brown clustering

1. Form estimate $\widehat{\boldsymbol{B}}$ of $\boldsymbol{B}$ matrix, e.g.,

$$\widehat{B}_{x,y} := \sum_{t=1}^{n-1} \widehat{\Pr}(X_t = x, X_{t+1} = y),$$

   and compute its rank-$|C|$ thin SVD $\widehat{\boldsymbol{U}}\widehat{\boldsymbol{S}}\widehat{\boldsymbol{V}}^\top$.
2. For each $x \in V$, let $\boldsymbol{q}_x$ be the corresponding row in $\widehat{\boldsymbol{U}}$, normalized to have unit length.
3. Apply agglomerative clustering (e.g., average-linkage) to vectors $\{\boldsymbol{q}_x : x \in V\}$.

Output: a *hierarchy* of word classes.

Spectral algorithm for Brown clustering

Spectral algorithm for Brown clustering

1. Form estimate $\widehat{\boldsymbol{B}}$ of $\boldsymbol{B}$ matrix, e.g.,

$$\widehat{B}_{x,y} := \sum_{t=1}^{n-1} \widehat{\Pr}(X_t = x, X_{t+1} = y),$$

   and compute its rank-$|C|$ thin SVD $\widehat{\boldsymbol{U}}\widehat{\boldsymbol{S}}\widehat{\boldsymbol{V}}^\top$.

2. For each $x \in V$, let $\boldsymbol{q}_x$ be the corresponding row in $\widehat{\boldsymbol{U}}$, normalized to have unit length.
3. Apply agglomerative clustering (e.g., average-linkage) to vectors $\{\boldsymbol{q}_x : x \in V\}$.

Output: a *hierarchy* of word classes.

Bonus: Main computational bottleneck (SVD) is a well-studied numerical linear algebra problem with highly-optimized solutions.

Improvements

Context X_{t+1} is "linguistic context" for X_t.
Can also use richer context
e.g., $(X_{t-2}, X_{t-1}, X_{t+1}, X_{t+2})$
(two words before, two words after).

Improvements

Context X_{t+1} is "linguistic context" for X_t.

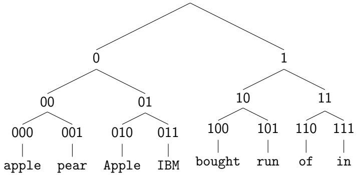
Can also use richer context
e.g., $(X_{t-2}, X_{t-1}, X_{t+1}, X_{t+2})$
(two words before, two words after).

Transforms Main Theorem holds even if we apply certain linear transformations to $\boldsymbol{B}$.

Does not change core structural properties,
but may improve conditioning.

Empirical study

Both Brown clustering and spectral algorithm provide (hierarchy of) word classes.



Questions:

1. How does spectral algorithm compare to Brown clustering on Brown clustering objective ($\widehat{\mathrm{MI}}$ between adjacent classes)?
2. How does spectral algorithm compare to Brown clustering in utility of lexical representations?

Question 1: Brown clustering objective

Data: RCV1 news articles (205M tokens).

Method: Compare Brown clustering with Spectral algorithm, both with $|C| = 1000$ classes.

Algorithm	$\|V\|$	$\widehat{\text{MI}}$	Time
Spectral	50K	1.48	0.37h
	300K	1.54	2.07h
Brown	50K	1.52	3.62h
	300K	1.60	22.33h

Question 2: Utility of lexical representations

Data: News articles for CoNLL 2003 Named Entity Recognition shared task.

Method: Using $|C| = 1000$ lexical representations from RCV1, with Perceptron + greedy decoding (Ratinov and Roth, 2009).
(Standard semi-supervised approach to this NLP problem.)

Algorithm	$\lvert V \rvert$	dev F1	test F1
Baseline		90.03	84.39
Spectral	50K	92.00	86.72
	300K	92.31	87.76
Brown	50K	92.00	88.56
	300K	92.68	88.76

(There is known discrepancy between dev & test sets here.)

Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to $10\times$ speed-up).

Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to $10\times$ speed-up).
- Spectral algorithm lags Brown clustering in terms of Brown clustering objective ($\widehat{\mathsf{MI}}$).

Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to $10\times$ speed-up).
- Spectral algorithm lags Brown clustering in terms of Brown clustering objective ($\widehat{\mathsf{MI}}$).
- Both algorithms provide lexical representations that deliver comparable improvements over baseline.

Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to $10\times$ speed-up).
- Spectral algorithm lags Brown clustering in terms of Brown clustering objective ($\widehat{\text{MI}}$).
- Both algorithms provide lexical representations that deliver comparable improvements over baseline.

Major limitation: Hard-clustering forces a class to capture all senses of any member word.

Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to $10\times$ speed-up).
- Spectral algorithm lags Brown clustering in terms of Brown clustering objective ($\widehat{\mathrm{MI}}$).
- Both algorithms provide lexical representations that deliver comparable improvements over baseline.

Major limitation: Hard-clustering forces a class to capture all senses of any member word.

Possible fix: Skip the clustering step! Directly use representation given by left singular vectors of $\widehat{\boldsymbol{B}}$.

3. Dealing with noise heteroskedasticity

Motivation

- Main estimation task in spectral algorithm is estimating word/context pairs frequencies $\boldsymbol{B}$
  (more specifically, the left singular vectors of $\boldsymbol{B}$).
- How can we do better on this estimation task?
- **Challenge**: many word/context pairs have very different frequencies, and hence very different "estimation noise variance".

Basic spectral algorithm

Simplified setting: word is X, context is Y.

Basic spectral algorithm

Simplified setting: word is X, context is Y.

Basic spectral algorithm:

- Use raw co-occurrence counts from N sentences

$$\widehat{B}_{x,y} := \#(X = x,\, Y = y)$$

(ignoring normalization).

- Decompose into low-rank factors using SVD, i.e., minimize

$$\min_{\substack{\boldsymbol{L} \in \mathbb{R}^{V \times C}, \\ \boldsymbol{R} \in \mathbb{R}^{V \times C}}} \|\boldsymbol{L}\boldsymbol{R}^\top - \widehat{\boldsymbol{B}}\|_F^2 \,.$$

Possible improvement

Since "noise" $\widehat{\boldsymbol{B}} - \boldsymbol{B}$ is highly heteroskedastic, could be better to minimize variance-normalized squared error

$$\min_{\substack{\boldsymbol{L} \in \mathbb{R}^{V \times C}, \\ \boldsymbol{R} \in \mathbb{R}^{V \times C}}} \sum_{x,y} \frac{1}{\operatorname{var}(\widehat{B}_{x,y})} \left((\boldsymbol{L}\boldsymbol{R}^{\top})_{x,y} - \widehat{B}_{x,y} \right)^2 .$$

(C.f. weighted least squares.)

Possible improvement

Since "noise" $\widehat{\boldsymbol{B}} - \boldsymbol{B}$ is highly heteroskedastic, could be better to minimize variance-normalized squared error

$$\min_{\substack{\boldsymbol{L} \in \mathbb{R}^{V \times C}, \\ \boldsymbol{R} \in \mathbb{R}^{V \times C}}} \sum_{x,y} \frac{1}{\operatorname{var}(\widehat{B}_{x,y})} \left((\boldsymbol{L}\boldsymbol{R}^\top)_{x,y} - \widehat{B}_{x,y} \right)^2 .$$

(C.f. weighted least squares.)

However, weighted objective is hard to minimize (Srebro *et al*, 2003).

A statistical trick

Square-root trick: Instead of using $\widehat{\boldsymbol{B}}$, use $\sqrt{\widehat{\boldsymbol{B}}}$ (*element-wise* square-root of $\widehat{\boldsymbol{B}}$).

A statistical trick

Square-root trick: Instead of using $\widehat{\boldsymbol{B}}$, use $\sqrt{\widehat{\boldsymbol{B}}}$ (*element-wise* square-root of $\widehat{\boldsymbol{B}}$).

Asymptotic justification:

- **Poisson approximation**: when $p_{x,y} := \Pr(X = x, Y = y)$ is small compared to $1/N$, approximately have

$$\widehat{B}_{x,y} \sim \mathrm{Poi}(N \cdot p_{x,y}) \,.$$

A statistical trick

Square-root trick: Instead of using $\widehat{\boldsymbol{B}}$, use $\sqrt{\widehat{\boldsymbol{B}}}$ (*element-wise* square-root of $\widehat{\boldsymbol{B}}$).

Asymptotic justification:

- **Poisson approximation**: when $p_{x,y} := \Pr(X = x, Y = y)$ is small compared to $1/N$, approximately have

$$\widehat{B}_{x,y} \sim \mathrm{Poi}(N \cdot p_{x,y}) \, .$$

- **Variance stabilization**: As $N \to \infty$,

$$\mathrm{var}\left(\sqrt{\widehat{B}_{x,y}}\right) \to 1/4$$

(Bartlett, 1936; Anscombe, 1948).

Variance stabilization

A heuristic derivation via delta method:

For $g(x) := \sqrt{x}$ and $X \sim \mathrm{Poi}(\lambda)$,

$$g(X) \quad \approx \quad g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X))$$

Variance stabilization

A heuristic derivation via delta method:
For $g(x) := \sqrt{x}$ and $X \sim \text{Poi}(\lambda)$,

$$\begin{aligned} g(X) &\approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X)) \\ &= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda)\,. \end{aligned}$$

Variance stabilization

A heuristic derivation via delta method:
For $g(x) := \sqrt{x}$ and $X \sim \mathrm{Poi}(\lambda)$,

$$\begin{aligned} g(X) &\approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X)) \\ &= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda)\,. \end{aligned}$$

Therefore

$$\mathrm{var}(g(X)) \approx \left(\frac{1}{2\sqrt{\lambda}}\right)^2 \cdot \mathrm{var}(X)$$

Variance stabilization

A heuristic derivation via delta method:
For $g(x) := \sqrt{x}$ and $X \sim \mathrm{Poi}(\lambda)$,

$$\begin{aligned} g(X) &\approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X)) \\ &= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda)\,. \end{aligned}$$

Therefore

$$\begin{aligned} \mathrm{var}(g(X)) &\approx \left(\frac{1}{2\sqrt{\lambda}}\right)^2 \cdot \mathrm{var}(X) \\ &= \frac{1}{4\lambda} \cdot \lambda \end{aligned}$$

Variance stabilization

A heuristic derivation via delta method:
For $g(x) := \sqrt{x}$ and $X \sim \mathrm{Poi}(\lambda)$,

$$\begin{aligned} g(X) &\approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X)) \\ &= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda)\,. \end{aligned}$$

Therefore

$$\begin{aligned} \mathrm{var}(g(X)) &\approx \left(\frac{1}{2\sqrt{\lambda}}\right)^2 \cdot \mathrm{var}(X) \\ &= \frac{1}{4\lambda} \cdot \lambda \;=\; \frac{1}{4}\,. \end{aligned}$$

Variance stabilization

A heuristic derivation via delta method:

For $g(x) := \sqrt{x}$ and $X \sim \mathrm{Poi}(\lambda)$,

$$\begin{aligned} g(X) &\approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X)) \\ &= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda)\,. \end{aligned}$$

Therefore

$$\begin{aligned} \mathrm{var}(g(X)) &\approx \left(\frac{1}{2\sqrt{\lambda}}\right)^2 \cdot \mathrm{var}(X) \\ &= \frac{1}{4\lambda} \cdot \lambda \; = \; \frac{1}{4}\,. \end{aligned}$$

So asymptotically, don't need variance normalization.

Variance stabilization

A heuristic derivation via delta method:

For $g(x) := \sqrt{x}$ and $X \sim \mathrm{Poi}(\lambda)$,

$$\begin{aligned} g(X) &\approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X)) \\ &= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda) \,. \end{aligned}$$

Therefore

$$\begin{aligned} \mathrm{var}(g(X)) &\approx \left(\frac{1}{2\sqrt{\lambda}}\right)^2 \cdot \mathrm{var}(X) \\ &= \frac{1}{4\lambda} \cdot \lambda \;=\; \frac{1}{4} \,. \end{aligned}$$

So asymptotically, don't need variance normalization.

Moreover, using $\sqrt{\widehat{\boldsymbol{B}}}$ make senses in the Brown model:

Left singluar vectors of $\sqrt{\boldsymbol{B}}$ also reveal word classes, just like $\boldsymbol{B}$'s.

Empirical study

Question: Does the Brown word class model capture the same intrinsic qualities as other popular lexical representations?

Empirical study

Question: Does the Brown word class model capture the same intrinsic qualities as other popular lexical representations?

- **Synonyms**: How well do cosine similarities between lexical representations reflect human judgements?

Empirical study

Question: Does the Brown word class model capture the same intrinsic qualities as other popular lexical representations?

- **Synonyms**: How well do cosine similarities between lexical representations reflect human judgements?
- **Analogies**: How well do lexical representations provide answer to analogy problems like

  Canberra is to *Australia*, as *London* is to ______

  based on cosine similarities:

$$\arg\max_{x \in V} \langle \boldsymbol{q}_x, \boldsymbol{q}_{\text{Australia}} \rangle - \langle \boldsymbol{q}_x, \boldsymbol{q}_{\text{Canberra}} \rangle + \langle \boldsymbol{q}_x, \boldsymbol{q}_{\text{London}} \rangle \,.$$

Empirical study

Question: Does the Brown word class model capture the same intrinsic qualities as other popular lexical representations?

- **Synonyms**: How well do cosine similarities between lexical representations reflect human judgements?
- **Analogies**: How well do lexical representations provide answer to analogy problems like

  Canberra is to Australia, as London is to _______

  based on cosine similarities:

$$\arg\max_{x \in V} \langle \boldsymbol{q}_x, \boldsymbol{q}_{\text{Australia}} \rangle - \langle \boldsymbol{q}_x, \boldsymbol{q}_{\text{Canberra}} \rangle + \langle \boldsymbol{q}_x, \boldsymbol{q}_{\text{London}} \rangle .$$

Are these measures predictive of utility in extrinsic tasks?

Data sources

- **Training data**: English Wikipedia, 1.4B tokens.

Data sources

- **Training data**: English Wikipedia, 1.4B tokens.
- **Similarity tasks**: Agirre *et al*'s "WordSim353", Bruni *et al*'s "MEN Test Collection", and Stanford Rare Word Similarity Dataset: (5.4K word pairs + human assessments)

  Measure Pearson correlation with human assessments.

Data sources

- **Training data**: English Wikipedia, 1.4B tokens.
- **Similarity tasks**: Agirre *et al*'s "WordSim353", Bruni *et al*'s "MEN Test Collection", and Stanford Rare Word Similarity Dataset: (5.4K word pairs + human assessments)

  Measure Pearson correlation with human assessments.
- **Analogy tasks**: Microsoft (Mikolov-Yih-Zweig) dataset of "syntatic" analogies: (8000 questions)

  Google (Mikolov *et al*) dataset of "syntatic" and "semantic" analogies: 19544 questions

  Measure word prediction accuracy.

Results

Other methods (with same context $X_{t-2}, X_{t-1}, X_{t+1}, X_{t+2}$ as Spectral):

- Continous bag-of-words (Mikolov *et al*, 2013) in Word2Vec
- Skip-gram (Mikolov *et al*, 2013) in Word2Vec
- PPMI (Levy and Goldberg, 2014)
- Glove (Pennington, Socher, Manning, 2014)

Method	dimension = 500			dimension = 1000		
	SIM (corr)	MSFT (acc%)	GOOG (acc%)	SIM (corr)	MSFT (acc%)	GOOG (acc%)
Spectral	0.572	39.68	57.64			
Spectral$_{+\sqrt{}}$	0.655	68.38	74.17	0.650	66.08	76.38
CBOW	0.597	75.79	73.60	0.509	70.97	60.12
SKIP	0.642	81.08	78.73	0.641	79.98	83.35
PPMI	0.628	43.81	58.38	0.637	48.99	63.82
Glove	0.576	68.30	78.08	0.586	67.40	78.73

Utility in extrinsic tasks

Directly use vectors $\boldsymbol{q}_x$ as features in structured prediction for Named Entity Recognition (again, CoNLL 2003 shared task).

Method	30 dimensions		50 dimensions	
	dev F1	test F1	dev F1	test F1
Baseline	90.03	84.39	90.03	84.39
Brown	92.49	88.75	92.49	88.75
Spectral$_{+\sqrt{\ }}$	92.88	89.28	92.94	89.01
CBOW	92.44	88.34	92.83	89.21
SKIP	92.63	88.78	93.11	89.32
PPMI	92.25	89.27	92.53	89.37
Glove	91.49	87.16	91.58	86.80

(There is known discrepancy between dev & test sets here.)

Utility in extrinsic tasks

Directly use vectors $\boldsymbol{q}_x$ as features in structured prediction for Named Entity Recognition (again, CoNLL 2003 shared task).

Method	30 dimensions		50 dimensions	
	dev F1	test F1	dev F1	test F1
Baseline	90.03	84.39	90.03	84.39
Brown	92.49	88.75	92.49	88.75
Spectral$_{+\sqrt{}}$	92.88	89.28	92.94	89.01
CBOW	92.44	88.34	92.83	89.21
SKIP	92.63	88.78	93.11	89.32
PPMI	92.25	89.27	92.53	89.37
Glove	91.49	87.16	91.58	86.80

(There is known discrepancy between dev & test sets here.)

All improve over baseline; **Spectral**$_{+\sqrt{}}$ is computationally cheapest.

Observations

- Spectral performs well on similarity tasks, less competitive on analogy tasks.
- Poor analogy performance doesn't seem to hurt much for extrinsic NER task.

Observations

- Spectral performs well on similarity tasks, less competitive on analogy tasks.
- Poor analogy performance doesn't seem to hurt much for extrinsic NER task.

Question: Which extrinsic tasks are analogy-adept lexical representations especially good for?

4. Unsupervised learning

Capturing linguistic structure without supervision

What linguistic structure is captured by HMM?

Capturing linguistic structure without supervision

What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)

Capturing linguistic structure without supervision

What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)
- **Test**: Do word classes correspond to parts-of-speech?

  Learn word class model, then measure "many-to-one accuracy", using *true* labels of words (e.g., from a dictionary).

Capturing linguistic structure without supervision

What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)
- **Test**: Do word classes correspond to parts-of-speech?

  Learn word class model, then measure "many-to-one accuracy", using *true* labels of words (e.g., from a dictionary).
- **Folklore**: maximum likelihood HMM is unlikely to yield states that correspond to parts-of-speech.

Capturing linguistic structure without supervision

What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)
- **Test**: Do word classes correspond to parts-of-speech?

  Learn word class model, then measure "many-to-one accuracy", using *true* labels of words (e.g., from a dictionary).
- **Folklore**: maximum likelihood HMM is unlikely to yield states that correspond to parts-of-speech.

  Evidence: running (unsupervised) EM, initialized at HMM learned with supervision, only makes things worse.

Capturing linguistic structure without supervision

What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)

- **Test**: Do word classes correspond to parts-of-speech?

  Learn word class model, then measure "many-to-one accuracy", using *true* labels of words (e.g., from a dictionary).

- **Folklore**: maximum likelihood HMM is unlikely to yield states that correspond to parts-of-speech.

  Evidence: running (unsupervised) EM, initialized at HMM learned with supervision, only makes things worse.

  Upshot: Do not use likelihood to test the hypothesis.

Linguistic context

Instead of likelihood, exploit linguistic context.

- Find HMM consistent with linguistic context (e.g., surrounding words) and features (e.g., spelling features).

Linguistic context

Instead of likelihood, exploit linguistic context.

- Find HMM consistent with linguistic context (e.g., surrounding words) and features (e.g., spelling features).

Our approach:

- Use spectral algorithm to derive lexical representation vectors.
- Apply farthest-first traversal to these vectors to pick "anchors".
- Use Bayes' rule + convex optimization to estimate HMM parameters (previously proposed by Arora-Ge-Moitra, 2012).

Some results

Data from universal treebank, 12 POS tag types

Many-to-one prediction accuracy

Method	de	en	es	fr	id	it	ja
E-M	45.5	59.8	60.6	60.1	49.6	51.5	59.5
Brown	60.0	62.9	67.4	66.4	59.3	66.1	60.3
$\text{Spectral}_{+\sqrt{}}$	61.1	66.1	69.0	68.2	63.7	60.4	65.3
$\text{Spectral}_{+\sqrt{}}+f$	63.4	71.4	74.3	71.9	67.3	60.2	69.4
Log-linear	67.5	62.4	67.1	62.1	61.3	52.9	78.2

- $\text{Spectral}_{+\sqrt{}}$ = just use prev/next words context.
- $\text{Spectral}_{+\sqrt{}}+f$ = also uses spelling features.
- Log-linear (Berg-Kirkpatrick *et al*, 2010): not a HMM

Final remarks

- Yet more confirmation that linguistic context is very powerful:

Final remarks

- Yet more confirmation that linguistic context is very powerful:

  "We" already knew the information was there; just need algorithmic/statistical techniques to fully exploit it.

Final remarks

- Yet more confirmation that linguistic context is very powerful:

  "We" already knew the information was there; just need algorithmic/statistical techniques to fully exploit it.

- Brown *et al* word class model is surprisingly simple — an obviously "wrong" model, but captures a lot of useful structure.

Final remarks

- Yet more confirmation that linguistic context is very powerful:

  "We" already knew the information was there; just need algorithmic/statistical techniques to fully exploit it.

- Brown *et al* word class model is surprisingly simple — an obviously "wrong" model, but captures a lot of useful structure.

- Unclear what is the "right" intrinsic evaluation of lexical representations.

Final remarks

- Yet more confirmation that linguistic context is very powerful:

  "We" already knew the information was there; just need algorithmic/statistical techniques to fully exploit it.

- Brown *et al* word class model is surprisingly simple — an obviously "wrong" model, but captures a lot of useful structure.

- Unclear what is the "right" intrinsic evaluation of lexical representations.

Thank you!

Connection to anchor word assumption

Anchor word assumption (Arora, Ge, Moitra, 2012) is strictly weaker than assumption in Brown word class model.

- For each hidden state $h \in C$, there is an "anchor" word $x \in V$ satisfying

$$O_{x,g} = 0 \text{ for all } g \neq h \,.$$

Connection to anchor word assumption

Anchor word assumption (Arora, Ge, Moitra, 2012) is strictly weaker than assumption in Brown word class model.

- For each hidden state $h \in C$, there is an "anchor" word $x \in V$ satisfying
$$O_{x,g} = 0 \text{ for all } g \neq h\,.$$
- For comparison, Brown model word class assumption requires *every word* to be an "anchor".

Connection to anchor word assumption

Anchor word assumption (Arora, Ge, Moitra, 2012) is strictly weaker than assumption in Brown word class model.

- For each hidden state $h \in C$, there is an "anchor" word $x \in V$ satisfying
$$O_{x,g} = 0 \text{ for all } g \neq h\,.$$
- For comparison, Brown model word class assumption requires *every word* to be an "anchor".

Stronger assumption can motivate different algorithmic choices (e.g., clustering normalized rows of left singular vector matrix).

Effect of representation

Data from English treebank, 12 POS tag types

Method	dev acc
Anchor	53.4
Anchor+CCA	57.0
Anchor+Rand	48.2
Spectral$_{+\sqrt{\ }}$	**66.1**

- Anchor = Arora-Ge-Moitra "conditional probability" representation.
- Anchor-CCA = same as Arora-Ge-Moitra, except apply CCA projection to right-hand side (Cohen-Collins, 2014).
- Anchor-Rand = same as Arora-Ge-Moitra, except apply random projection to right-hand side (Ding *et al*, 2013).